In the Claims

1. **(Currently Amended)** A method comprising:

providing information relating to a business application in a server system, comprising

receiving a request, wherein

the request is configured to cause the business application to execute a command of the business application,

the request comprises an indication of a user interface element to be returned, **[[and]]**

the command is configured to be defined by a data structure comprising

an execute element,

a command element, and

an argument element, and

the command **[[is]]** **element represents** a predefined query;

generating a data element by executing the command of the business application **~~to generate a data element~~**;

generating the user interface element to be returned in response to the request; and

sending a response comprising the user interface element and the data element, **wherein the user interface element and the data element are XML documents**.

2. **(Currently Amended)** The method of claim 1 wherein the **~~request~~** **argument element** indicates a type of user interface element to return.

3. **(Currently Amended)** The method of claim 1 wherein the **~~request~~** **argument element** indicates a type of user interface element to not return.

4. (Original) The method of claim 3 wherein the type of user interface element not to return is navigation data.

5. **(Currently Amended)** The method of claim 1 wherein the ~~request~~ **argument element** comprises an "SWEDataOnly" argument.

6. **(Currently Amended)** The method of claim 1 wherein the ~~request~~ **argument element** comprises an "SWEApplet" argument.

7. (Previously Presented) The method of claim 1 further comprising:

receiving a list of predefined queries in response to the request, wherein

the list of predefined queries comprises the predefined query.

8. **(Currently Amended)** A method in a server system for providing information relating to a business application, the method comprising:

providing transforms for transforming output of the business application, each transform having a name;

receiving a request to execute a command of a business application, wherein

the request is received from a client system,

the command is configured to be defined by a data structure comprising

an execute element,

a command element, and

an argument element,

the command **[[is]] element represents** a predefined query, and

the ~~request~~ **argument element** optionally indicates the name of a transform to be applied to the output of the business application;

generating a generated output by executing the command of the business application ~~to generate output~~ **, wherein the request is in XML format and the transforms are XSLT stylesheets**;

when the ~~request~~ **argument element** indicates the name of **[[a]] the** transform,

generating a transformed output by applying the ~~provided~~ transform ~~with the indicated name~~ to the generated output ~~to generate transformed output~~, and

sending **the transformed output** to the client system ~~the transformed output~~; and

~~when the request does not indicate the name of a transform~~ otherwise, sending the generated output to the client system ~~the generated output~~.

9. **(Cancelled)**

10. (Previously Presented) The method of claim 8 wherein the request comprises an "SWEStyleSheet" argument.

11. **(Currently Amended)** A method in a server system for providing information relating to a business application, the method comprising:

providing a default format for output of the business application;

receiving a request to execute a command of a business application,

the request is received from a client system,

the command is configured to be defined by a data structure comprising

an execute element,

a command element, and

an argument element,

the command **[[is]]** element represents a predefined query, and

the ~~request~~ argument element optionally indicates a user agent format or a client-specified format for the output of the business application;

selecting a format giving preference in the following order: the client-specified format, the user-agent format, and the default format;

generating a generated output by executing the command of the business application ~~to generate output~~ , wherein the request is in XML format;

sending the generated output in the selected format to the client system ~~the generated output in the selected format~~.

12. (Original) The method of claim 11 wherein the user-agent format is selected over the default format in accordance with a predefined preference of formats.

13. (Original) The method of claim 11 wherein the user-agent format is based on type of user agent specified in the request.

14. (Original) The method of claim 13 wherein the type of user agent specifies a type of browser.

15. (Original) The method of claim 11 wherein the formats are a markup language.

16. (Original) The method of claim 15 wherein one of the formats is HTML.

17. (Original) The method of claim 15 wherein one of the formats is XML.

18. (Original) The method of claim 15 wherein one of the formats is WML.

19. (Previously Presented) The method of claim 11 wherein the request comprises an "SWESetMarkup" argument that specifies the client-specified format.

20. (**Currently Amended**) A computer-readable medium containing:

instructions, executable on a computer system, configured to execute a command of a business application; and

a data structure defining ~~**an inbound**~~ **the** command**, wherein the command is inbound** to a web server **and the web server is configured to execute on the computer system**, the data structure comprising**[[:]]**

an execute element having a path attribute indicating a location of an object manager**[[;]]** ,

a command element nested within the execute element and having a value attribute indicating a name of **[[a]] the** command ~~**to execute**~~, wherein the command element represents a predefined query**[[;]]** , and

one or more argument elements nested within the command element, each argument element having a name attribute indicating a name of an argument for the ~~**named**~~ command, the one or more argument elements being from a set of argument elements comprising an argument element for indicating a response markup format, an argument element for indicating whether the response should include user interface elements,

and an arguments element identifying a transform to be applied to output, **wherein the data structure is an XML document**.

21. **(Cancelled)**

22. (Previously Presented) The computer-readable medium of claim 20 wherein zero or more occurrences of the command element are nested within the execute element.

23. (Original) The computer-readable medium of claim 20 wherein only one command element is nested within the execute element.

24. **(Currently Amended)** A computer-readable medium containing:

instructions, executable on a computer system, configured to execute a command of a business application; and

a data structure defining ~~an outbound~~ **the** command, **wherein the command is outbound** to a web server **and the web server is configured to execute on the computer system**, the data structure comprising[[:]]

an application element having a name attribute[[;]] ,

a navigation element nested within the application element, having a name attribute, and having sub-elements from a set comprising a menu element, tool bar element, screen bar element, thread bar element, view bar element, and page item element[[;]] ,

a predefined query bar element nested within the application element and each having a name attribute[[;]] , and

one or more elements from the set of elements comprising a screen element, an applet element, and a form element, the one or more elements being nested within the application element and each having a name attribute, **wherein the data structure is an XML document**.

25. **(Currently Amended)** A method in a server system for providing information relating to a business application, the method comprising:

receiving a request to execute a command of a business application, wherein

the request is received from a client system,

the command is configured to be defined by a data structure comprising

an execute element,

a command element, and

an argument element,

the command **[[is]]** **element represents** a predefined query, and

the ~~request~~ **argument element** indicates a user interface element and a data element to be returned as results of execution of the command;

generating the data element by executing the command ~~to generate the data element~~ **, wherein the user interface element and the data element are XML documents;**

when the ~~request~~ **argument element** indicates to return at least one user interface element,

generating the at least one user interface element to be returned; and

sending **a first response** to the client system ~~a response that~~ **, wherein the first response** comprises the ~~generated~~ **at least one** user interface element and the ~~generated~~ data element; and

~~when the request indicates to not return the at least one user interface element~~ **otherwise**, sending **a second response** to the client system ~~a response comprising~~ **, wherein the second response comprises** the ~~generated~~ data element ~~without~~ **and the second response does not include** the user interface element.

26. (Original) The method of claim 25 wherein the request indicates a type of user interface element to return.

27. (Original) The method of claim 25 wherein the request indicates a type of user interface element to not return.

28. (Original) The method of claim 27 wherein the type of user interface element not to return is navigation data.

29. (Previously Presented) The method of claim 25 wherein the request comprises an "SWEDataOnly" argument.

30. (Previously Presented) The method of claim 25 wherein the request comprises an "SWEApplet" argument.

31. (Previously Presented) The method of claim 25 further comprising:

receiving a list of predefined queries in response to the request, wherein the list of predefined queries comprises the predefined query.